

REMARKS/ARGUMENTS

Request for Continued Examination:

5 The applicant respectfully requests continued examination of the above-indicated application as per 37 CFR 1.114.

1. Introduction

This is a full and timely response supplemental response to the Office Action of March 24, 2008. New claims 20-21 are added. No new material has been introduced.
10 Reconsideration of the application is respectfully requested.

2. Background

Claims 1-2, 7, and 17-19 are rejected under 35 U.S.C 103(a) as being unpatentable over Schieve (US 5463766) in view of Sanchez (US 6477666). Claims
15 3-4 and 8 are rejected under 35 U.S.C 103(a) as being unpatentable over Schieve in view of Sanchez and in further view of Phillips (US 5321828).

3. Response

The Examiner's position is that Schieve does not explicitly disclose resetting a
20 parameter to simulate the peripheral device being in the error state throughout execution of the event corresponding to the diagnosis code, however, Sanchez in the same analogous art of testing the computer program about reliable and proper handling of various faults under various conditions, discloses a method to simulate the error state throughout execution, and thereby rejects the independent claims 1 and 18

of the present application.

The present application provides a method for program debugging to allow users to debug without destroying any hardware. Therefore, when a firmware designer debugs a firmware program, he or she does not have to destroy a peripheral just to know if the
5 firmware program successfully detects the error state and executes the corresponding function under the condition that the peripheral is in the error state.

However, in Schieve's application, according to column 8, lines 21-25, "Next, execution proceeds to block 430, wherein the boot loading routine has read the rudimentary image directory and generates as a function of the peripherals detected and
10 the contents of that directory, an options screen" (emphasis added). Therefore, Schieve only teaches "testing" of the peripherals that are detected, and detection is done in step 410 (Fig.4), long before any test options become available (Fig.4, steps 430 and 440). In other words, by utilizing Schieve's application even as suggested by the Examiner, a user can only know if existing peripherals "pass the test" (Fig.4). Peripherals that are not
15 detected are not tested. On the other hand, present claim 1 includes limitations of processing both the general processing path and the error processing path for each event, regardless if the peripheral device being tested actually exists, something Schieve cannot do.

Furthermore, it is respectfully disagreed that Schieve's application and Sanchez's
20 application are should be combined as suggested. In fact, designing a computer system generally needs three kinds of designers/engineers: hardware designer, firmware designer, and software designer. The hardware designer designs the circuits and power of the computer system, the firmware designer designs the basic functions, drivers for the

circuits, and defines the basic interactions between the circuits by assembly programming, and after the debugging of the hardware and the firmware phases are done, then the operating system (for example, Windows) is loaded into the computer system. Since the operating system is loaded into the computer system, the software designer starts to

5 design software applications such as JAVA programming on the operating system.

Therefore, during the phases of firmware designing/debugging, the design environment is very difficult and very different from the design environment of the software because no operating system is loaded, which means no high-level commands can be used for debugging. The firmware designer can only use few simple commands to design and

10 debug other than the complex commands on the operating system, and this is very well known to those skilled in firmware designing. That is the reason and the motivation of the present application providing a method for debugging with destroying any peripherals. Therefore, a firmware designer cannot achieve the present application by only the teaching from Sanchez's application because Sanchez's application provides teaching in

15 the software art, not in the firmware art, and vice versa. For the reasons described above, Schieve's application and Sanchez's application cannot be reasonably combined.

Additionally, the Examiner does not explain how and why it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Sanchez's fault injection method to simulate the error state of peripheral devices in

20 Schieve, especially how to test both error and non-error code routines for peripheral devices that are not detected. According to Examination Guidelines for determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in KSR International Co. v. Teleflex Inc, "Rejections on obviousness cannot be sustained by mere

conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” Therefore, the applicant respectfully reminds the Examiner that rejections found in this OA are not supported by "substantial evidence" as required under the Administrative Procedure Act (APA). The applicant preserves all rights under the APA in appeals within the PTO as well as the judiciary.

Therefore, claim 1 of the present application is patentable over Schieve in view of Sanchez and should be allowable. Since claims 2-4, 6-8, and 17 are dependent on claim 1, claims 2-4, 6-8 and 17 should be allowable if claim 1 is allowable.

Claim 18 of the present invention is patentable over Schieve in view of Sanchez and should be allowable for the same reason above.

4. New claims

Please accept for consideration new independent claim 20 and claim 21 dependent thereon. Both claims are supported at least by paragraphs [0031] and [0033], and no new material has been introduced.

5. Summary

Schieve only teaches “testing” of the peripherals that are detected while present claim 1 includes limitations of processing both the general processing path and the error processing path for each event, regardless if the device exists or not.

Furthermore, it is respectfully asserted that Schieve’s application and Sanchez’s application are should not be combined as suggested at least because a firmware designer

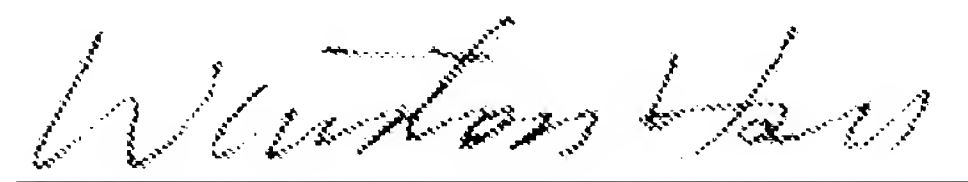
cannot achieve the present application by only the teaching from Sanchez's application since Sanchez's application provides teaching in the software art, not in the firmware art, and vice versa.

5 Additionally, the Examiner does not explain how and why it would have been obvious to use Sanchez's fault injection method to test both error and non-error code routines for peripheral devices that are not detected in Schieve.

Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

10

Sincerely yours,



Date: 09.22.2008

Winston Hsu, Patent Agent No. 41,526

15 P.O. BOX 506, Merrifield, VA 22116, U.S.A.

Voice Mail: 302-729-1562

Facsimile: 806-498-6673

e-mail : winstonhsu@naipo.com

20 Note: Please leave a message in my voice mail if you need to talk to me. (The time in D.C. is 12 hours behind the Taiwan time, i.e. 9 AM in D.C. = 9 PM in Taiwan.)